

Design and Implementation of High Speed UART Based on DMA

Feng Zang and Honghai Niu
NR Electric Co., Ltd. Nanjing, 211102, China

Abstract—When using DMA to receive messages, it is necessary to specify the number of bytes needed to be sent to the DMA controller. When the DMA controller receives the specified bytes, DMA transmission interruption is generated. But in practical applications, the number of bytes received by UART is often unfixed. In this paper, data transmission through DMA has a fixed-length problem. We design and implement a way to solve the problem through external interruption and timer matching. This scheme takes into account the advantages of DMA transmission without the need of CPU intervention, and makes up for the defect that it cannot transmit the indefinite length message. The external interruption is generated by an external pin connected to the UART RXD pin, which is associated with the input event of the timer. Set timer as Retrigger mode. After receiving EVENT, the accumulative time of the timer is cleared. When the timer fails to receive EVENT within a certain time, it will go out of time and make message processing in the timeout interrupt service program. Due to the use of timeout interruption for message processing, rather than the fixed length interrupt processing of DMA, it can achieve the reception and processing of indefinite length message, and meet the demand of UART to transmit indefinite long packets.

Keywords—DMA; indefinite length message; external interrupt; UART; timer; timeout interrupt

I. INTRODUCTION

UART (Universal Asynchronous Receiver/Transmitter) is widely applied in industrial control & communication. It is designed to implement communication between main controller & auxiliary devices [1] [2]. Generally speaking, 2 modes are applied in UART data transmission: interrupt mode and DMA (Direct Memory Access) mode [3] [4].

In interrupt mode, CPU usage is high, mass high-priority data transmission will occupy most of the CPU resources, so that low-priority processes must wait for a very long time.

To solve this problem, DMA mode can be considered: CPU control can be skipped by application of DMA, so the data can be acquired from the memory directly [5] [6] [7] [8].

With identification of address space, DMA can provide high-speed data transmission between peripherals and memories [9]. Therefore, reliable unidirectional data transmission between source address & destination address of single channel is provided [10]. However, only messages with definite length can be transmitted in DMA mode, i.e.: short data message will be complemented to the pre-defined definite length with invalid data. Therefore, received byte number must be specified for the DMA controller, but received data length of

the serial port is indefinite in engineering application, so DMA mode is not suitable to receive data for such situation.

In this paper, a new method based on DMA is designed to solve this problem.

II. METHODOLOGY

A. Overall Design

A new method based on DMA is provided to receive indefinite length high speed UART message. Detailed steps are listed as below:

- (1) Set the serial port to DMA reception mode.
- (2) Set a GPIO (General-Purpose Input/Output) pin as the EIC (External Interrupt Controller). In the paper, PC14 pin is configured as the GPIO. Connect this pin to RXD pin (Receiving pin) of the serial port to generate external interrupt INT14.
- (3) Configure the external pin to the falling edge mode, i.e.: generate external interrupt upon falling edge. The timer counter TC0 will be enabled upon the first falling edge.
- (4) Configure the external interrupt to generate EVENT that is related with TC0.
- (5) Relate input EVENT of TC0 to the EVENT generated by the external pin. Set TC0 to "Retrigger" mode.
- (6) Set timeout duration for TC0. In this paper, the timeout duration is 20 μ s. TC0 will be disabled after the timeout duration, and message processing will be started. DMA reception mode will be enabled again to wait for the next frame message.

Design principle of this scheme is shown in the following "Figure I".

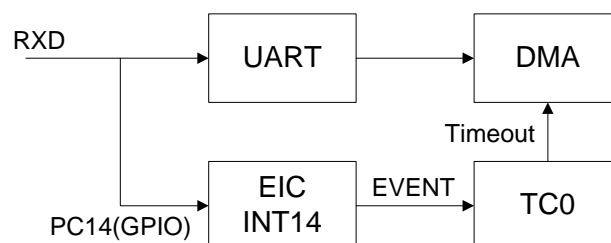


FIGURE I. DESIGN PRINCIPLE

B. Receive Data with UART DMA

In data transmission based on DMA, it is necessary to configure DMA controller first, and then the DMA channel source address of should be configured. This address is the register address of the UART reception data.

Finally, the DMA channel destination address should be configured. The address is the initial address of the buffer array to store the UART data. Moreover, the target address should be configured as self-increase mode.

C. External Interrupt

Set the GPIO pin (i.e.: PC14) as external interrupt mode, and then connect this pin to RXD pin. Configure the external pin to the falling edge mode, i.e.: generate external interrupt upon falling edge. The timer counter TC0 will be enabled upon the first falling edge.

Configure the external interrupt to generate EVENT that is related with TC0.

D. Timer

Set TC0 to "Retrigger" mode, i.e.: clear accumulated duration of the timer to 0 once EVENT is received. Because the UART data adopts low level as the start bit. When the UART is sending data, there will be a falling edge, so the timer TC0 will not overflow & generate timing interrupt. When the UART is not sending data, the continuous high level will lead to timing interrupt. In the timing interrupt service program, the timer will be closed, the DMA receiving channel will be reset to wait for next message, and then UART messages will be processed.

III. HARDWARE STRUCTURE

Atmel series ATSAME54N20A chip manufactured by Microchip Company is adopted in this scheme. Abundant peripheral interfaces are provided: 32 configurable DMA channels, 16 configurable external interrupt pins, 8 configurable UART serial ports [11]. Relevant hardware circuit diagram of UART is introduced in the following figure.

In this scheme, configurable UART pins are connected to pin A & pin B of the RS-485 chip. Receiving pin "RXD" are connected to external pin PC14, as shown in "Figure II".

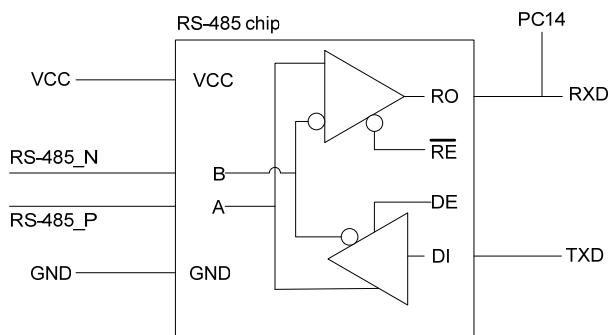


FIGURE II. UART CIRCUIT

IV. SOFTWARE PRINCIPLE

To ensure normal operation of the software, some modules of the chips must be initialized, e.g.: crystal oscillator, clock, etc.

Initialization for the following components are very important in this scheme:

1) UART:

Configure baud rate (5M bps in this example), Parity check bit, Stop bit, etc.

2) DMA channel:

Configure DMA channel for UART about source/destination address of DMA.

3) External interrupt:

Configure PC14 pin to external interrupt mode, and generate EVENT;

Configure start_timer_flag = 0. (It is the flag indicating first falling edge)

4) Timer:

Configure timeout setting for the timer, configure EVENT for the timer.

After initialization, PC14 pin will detect whether there is any falling edge.

If a falling edge is detected and start_timer_flag is 0, the timer will be started, and start_timer_flag will be set to 1.

Otherwise, the falling edge will generate EVENT, and the accumulated duration of the timer will be cleared to 0.

If no falling edge is generated, in case of timer timeout, the timeout interrupt service program will be invoked.

The timer will be closed first, and then the UART messages will be processed. DMA will be initialized again, and start_timer_flag will be set to 0.

Flow chart of UART data reception is shown in the following "Figure III".

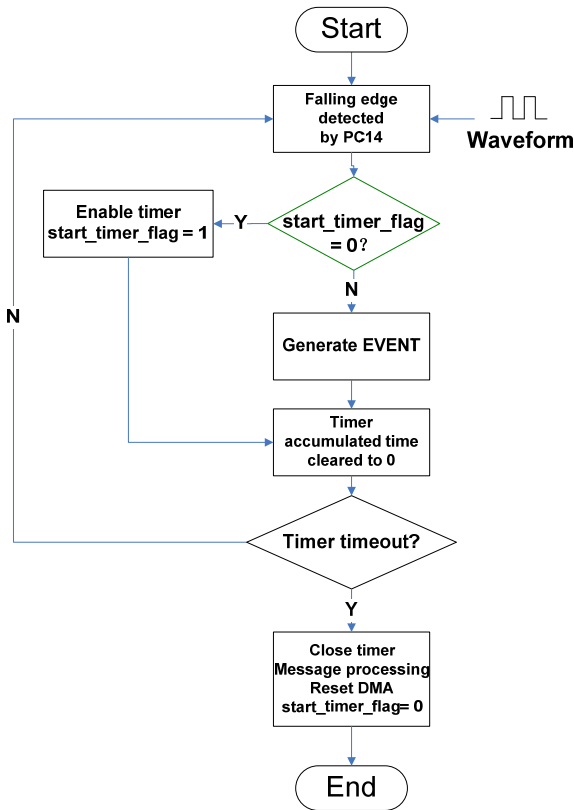


FIGURE III. FLOW CHART OF UART DATA RECEPTION

V. PERFORMANCE ANALYSIS

To explain this scheme in details, the following 4 figures in this chapter will be used for performance analysis.

In UART data reception, waveform of UART data is shown in "Figure IV".

Firstly, a low level is detected by the bus, it indicates data requiring to be sent. This low level is the start bit of the RXD pin. Because PC14 pin is connected to RXD, so the waveform of PC14 is the same as the RXD pin.

Waveform of external pin is shown in "Figure V". Because PC14 is configured as external interrupt mode, and the external interrupt is generated by the falling edge, when the start bit of RXD, or UART data bit changes from 1 to 0, a falling edge will be generated, so an external interrupt will be generated by PC14.

Waveform of external interrupt is shown in "Figure VI". The external interrupt will generate an EVENT. Because the EVENT is associated with the input event of TC0, when the EVENT is detected by TC0, TC0 will clear accumulated duration of the timer, and then restart timing from 0.

If no data to be received by UART, continuous high level of the bus will be detected, i.e.: RXD of UART remains "1". Now there is no EVENT to clear accumulated duration of the timer TC0, so the timer will overflow and generate an interrupt. Waveform of this process is shown in "Figure VII".

In this paper, the timeout setting of the timer is 20μs. In the timing interrupt service program, the DMA reception channel will be reset to wait for next frame, and then process UART messages.



FIGURE IV. WAVEFORM OF UART DATA

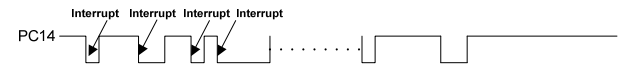


FIGURE V. WAVEFORM OF EXTERNAL PIN

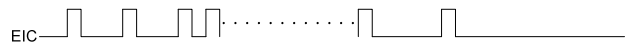


FIGURE VI. WAVEFORM OF EXTERNAL INTERRUPT



FIGURE VII. WAVEFORM OF TIMER

VI. TEST RESULTS AND ANALYSIS

A. Establish Test Environment

To prove correctness of above analysis, an 5M bps UART communication test environment is established as shown in "Figure VIII".

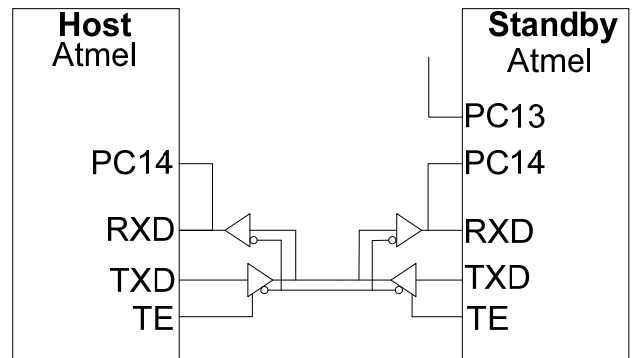


FIGURE VIII. TEST ENVIRONMENT

Test steps are listed as below:

- (1) Connect RS485 pin of main Atmel chip to that of standby Atmel directly.
- (2) Connect RXD pin to PC14 pin directly.
- (3) PC13 pin of standby Atmel is used to monitor TC0 accessing interrupt service program due to timeout. When timeout interrupt service program is enabled, high level will be enabled. Low level will be enabled by exiting from timeout interrupt service program. Connect this pin to the oscilloscope (OSC) to view waveform.

- (4) Configure receiving buffer of the DMA to 300 bytes. Check whether PC13 can output pulse signal by using the OSC.
- (5) Once the standby Atmel receives data to be processed, it will return 200 bytes to the main Atmel.

B. Analysis of Test Result

Test result analysis is shown in the above "Figure IX".



FIGURE IX. TEST RESULT

The following results can be concluded by analyzing waveforms in the Agilent OSC:

- (1) Yellow waveforms are the message waveforms of UART. Baud rate is 5M bps.
- (2) Sending 200 bytes data will cost 200 μ s. Minimum scale of the upper graph is 1ms.
- (3) Green waveforms are the waveforms of PC13 pin of standby Atmel. As shown in the upper sub-graph, when the main Atmel has sent data, it will access TC0 timeout interruption service program, and it will only access once.
- (4) The lower sub-graph is the partial enlarged graph of the upper graph. Minimum scale of the lower graph is 20 μ s.
- (5) As shown in the lower graph, when UART has sent data, with a time delay of about 30 μ s, timeout interruption service program will be invoked: UART timeout duration is 20 μ s, and upward/downward message switching duration of the interruption service program is about 10 μ s.

VII. CONCLUSION

In this paper, a new scheme is provided based on analysis of UART data reception via DMA. In this scheme, external

interrupt and timer cooperates, so the data transmission will not be influenced by CPU usage, and indefinite received data length of the UART can be processed. Therefore, CPU processing time is minimized, and UART data reception efficiency is improved. But an extra GPIO pin and an extra timer will be used, please apply this scheme cautiously if the chip resource is limited.

This scheme has been applied in many projects and proven effective.

REFERENCES

- [1] SUN Jinglong, WANG Yecheng and CHEN Rui, "Implementing High Speed UART on STM32F4xx with DMA Controller," *Heilongjiang Science and Technology Information*, no. 27, pp. 36-36, 2013.
- [2] Ding Xiaoli, Zhang Shoumin, "Application of Serial Communication in Electric Heating Fore-hearth Control System," *Process Automation Instrumentation*, vol. 33, no. 5, pp. 32-34, 2012.
- [3] SHAO Weiheng, GU Ang, and SUI Junjie, "Design of STM32-based Portable Constant Current Source," *Automation and Instrumentation*, vol. 29, no. 4, pp. 29-32, 2015.
- [4] Xing Lihua, Gao zhipeng, and Yuan Dedian, "High Speed UART Driver Based on Scatter DMA," *Microcontrollers and Embedded Systems*, no. 4, pp.40-42, 2010.
- [5] ZHANG Wei, HU Xiaonan, and WANG Ye, "Design of UART serial communication based on DMA controller," *Electronic Desing Engineering*, vol. 20, no. 6, pp. 138-140, 2012.
- [6] YANG Fuguang, LI Yibin, YIN Zhangfang, and LIU Wenjiang, "High efficiency UART communication based on DMA controller in ARM processor and its application," *Microcomputer Information*, vol. 24, no. 1, pp.161-163, 2008.
- [7] WANG Xiaojian, PAN Shunliang, SHEN Weiqun and SONG Zishan, "Implementing UART on the TMS320DM642 with the McBSP and EDMA," *Electronic Measurement Technology*, vol. 31, no. 2, pp. 103-105, 2008.
- [8] WU Guangzhou, ZHANG Ping, "Implementing a serial asynchronous communication on the TMS320C5510 with McBSp and DMA," *Electronic Measurement Technology*, vol. 29, no. 6, pp. 115-117, 2006.
- [9] WEI Lin and TIAN Bo, "Research UART communication on STM32F4 with DAM Controller," *Automation Application*, no. 8, pp. 92-93, 2016.
- [10] SONG Yan, "The Definitive Guide to the ARM Cortex-M3," *BeiHang University Press*. [M]
- [11] Microchip, SAM D5x/E5x Family datasheet[M].